This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1.   (Currently amended) A method for implementing a bimodal virtual device in a computer system, said method comprising:

receiving, by ~~configuring~~ the bimodal virtual device, a request, the bimodal virtual device configured to selectively operate ~~such that it selectively operates~~ with one or more virtual machines in two different modes, a first mode comprising a hardware mode during which the bimodal virtual device emulates a specific hardware device and is accessed by a virtual machine via a device driver that is configured to drive the specific hardware device by configuring emulated hardware components of the emulated specific hardware device, a processor requiring a first amount of cycles to emulate the hardware components of the emulated specific hardware device, and a second mode comprising an idealized mode, the idealized mode configured to send input/output instructions from a guest operating system to a host operating system, the processor requiring a second amount of cycles to send the input/output instructions, the second amount of cycles lower than the first ~~where the bimodal virtual device is optimized for a virtualized environment and wherein said virtual device operates without emulating the specific hardware device~~;

selecting an operating mode to service the request; and

executing an instruction in accordance with the selected operating mode.

2.   (Currently amended) The method of claim 1 wherein:

the bimodal virtual device is configured to ~~selectively~~ operate[[s]] in the hardware mode when a device driver interfacing with said bimodal virtual device has not been designed to interface with said bimodal virtual device operating in said second mode; and

the bimodal virtual device is configured to ~~selectively~~ operate[[s]] in the idealized mode when the driver interfacing with said bimodal virtual device has been designed to interface with said bimodal virtual device operating in said second mode.

3.   (Original) The method of claim 2 wherein the functionality of the second mode extends the functionality of the first mode.

4.   (Canceled)

5.   (Currently amended) The method of claim 4 wherein the functionality of the second mode <u>comprises the</u> functionality of the first mode.

6.   (Currently amended) The method of claim 4 wherein the functionality of the second mode <u>comprises</u> portions of the functionality of the first mode.

7.   (Currently amended) The method of claim 2 wherein the second mode is enabled throug<u>h at</u> least one bit in a virtual device register.

8.   (Currently amended) The method of claim 2 wherein the second mode is enabled throug<u>h at</u> least one bit in a register specifically created for utilization by one or more virtual <u>machines.</u>

9.   (Currently amended) The method of claim 2 wherein the second mode is enabled through <u>a</u> prescribed sequence of commands or data that change a value in at least one register

10.   (Currently amended) The method of claim 2 wherein
          the second mode is enabled through the use of a second mode driver instilled within a guest operating system environment; and
          if the second mode driver is not present, a first mode driver is instead enabled<u>.</u>

11.   (Currently amended) A computer system, said computer system comprising ~~a~~
~~bimodal virtual device that selectively operates as~~:

a processor operatively coupled to memory, the memory including instructions for a
bimodal virtual device;

the processor configured to effectuate a first mode of the bimodal virtual device, the
first mode operable to emulate hardware components of an emulated hardware device, the
processor requiring a first amount of processor cycles to effectuate the first mode;

the processor configured to effectuate a second mode of the bimodal virtual device,
the second mode optimized to send input/output instructions from a guest operating system to
a host operating system, the processor requiring a second amount of processor cycles to
effectuate the second mode, wherein the second amount of processor cycles is less than the
first

~~a hardware virtual device in a first mode; and~~

~~as an idealized virtual device in a second mode~~;

~~wherein in the first mode the bimodal virtual device emulates a real hardware device,~~
~~and in the second mode the bimodal virtual device functions as an abstract device that is a~~
~~same type of device as the real hardware device but is incompatible with software configured~~
~~to interact with the real hardware device.~~

12.   (Currently amended) The system of claim 11 wherein:

the bimodal virtual device is configured to ~~selectively~~ operate[[s]] when a
driver interfacing with said bimodal virtual device has not been designated to interface with
~~said~~ an abstract device; and

the bimodal virtual device is configured to ~~selectively~~ operate[[s]] in the
second mode when the driver interfacing with said bimodal virtual device has been
designated to interface with said bimodal virtual device operating in said second mode.

13.   (Original) The system of claim 12 wherein the functionality of the second
mode extends the functionality of the first mode.

14.   (Original) The system of claim 12 wherein the functionality of the second mode is independent of the functionality of the first mode.

15.   (Original) The system of claim 14 wherein the functionality of the second mode disables the functionality of the first mode.

16.   (Original) The system of claim 14 wherein the functionality of the second mode disables portions of the functionality of the first mode.

17.   (Original) The system of claim 12 wherein the second mode is enabled through the use of at least one bit in a virtual device register.

18.   (Original) The system of claim 12 wherein the second mode is enabled through the use of at least one bit in a register specifically created for utilization by one or more virtual devices.

19.   (Original) The system of claim 12 wherein the second mode is enabled through the use of a prescribed sequence of commands or data that change a value in at least one register.

20.   (Currently amended) The system of claim 12 wherein
        the second mode is enabled through the use of a second mode driver installed within a guest operating system environment[[; and]].

21.   (Currently amended) A computer system, said computer system comprising:

a processor operatively coupled to memory;

the processor configured to effectuate a virtual machine environment; and

the processor configured to effectuate a bimodal virtual device that is configured to operate ~~selectively operates~~ in either a first or second mode ~~as a hardware virtual device in a first mode~~;

the processor configured to operate in the first mode by executing instructions that effectuate emulated hardware components of the hardware virtual device, the processor requiring a first amount of processor cycles to effectuate the emulated hardware components;

the processor configured to operate in the second mode by executing instructions that send input/output instructions from a guest operating system to a host operating system, the processor requiring a second amount of processor cycles to send the input/output instructions, the second amount of processor cycles lower than the first ~~as an idealized virtual device in a second mode with said virtual machine environment, where in the first mode the bimodal virtual device emulates a real hardware device, and in the second mode the bimodal virtual device functions as an abstract device that is a same type of device as the real hardware device but is incompatible with software configured to interact with the real hardware device~~.

22.   (Previously presented) The system of claim 21 wherein:

the bimodal virtual device is configured to ~~selectively~~ operate[[s]] in the first mode when a driver interfacing with said bimodal virtual device has not been designated to interface with said bimodal device operating in said second mode; and

the bimodal virtual device is configured to ~~selectively~~ operate[[s]] in the second mode when the driver interfacing with said bimodal virtual device has been designated to interface with said bimodal virtual device operating in said second mode.

23.   (Original) The system of claim 22 wherein the functionality of the second mode extends the functionality of the first mode.

24.   (Original) The system of claim 22 wherein the functionality of the second mode is independent of the functionality of the first mode.

25.   (Original) The system of claim 24 wherein the functionality of the second mode disables the functionality of the first mode.

26.   (Original) The system of claim 24 wherein the functionality of the second mode disables portions of the functionality of the first mode.

27.   (Original) The system of claim 22 wherein the second mode is enabled through the use of at least one bit in a virtual device register.

28.   (Original) The system of claim 22 wherein the second mode is enabled through the use of at least one bit in a register specifically created for utilization by one or more virtual devices.

29.   (Original) The system of claim 22 wherein the second mode is enabled through the use of a prescribed sequence of commands or data that changes a value in at least one register.

30.   (Original) The system of claim 22 wherein

        the second mode is enabled through the use of a second mode driver installed within a guest operating system environment; and

        if the second mode driver is not present, a first mode driver is instead enabled.

31.   (Currently amended) A computer-readable storage medium comprising computer-readable instructions, said computer-readable instructions comprising:

instructions for a bimodal virtual device, the instructions for bimodal virtual device operable to configure a processor to operate in a first or a second mode,

the first mode emulating hardware components of a hardware device, the first mode requiring a first amount of processor cycles to emulate the hardware components;

the second mode configured to send input/output instructions from a guest operating system to a host operating system, the second mode requiring a second amount of processor cycles to send the input/output instructions, the second amount of processor cycles being less than the first to selectively operate as a hardware virtual device in a first mode and as an idealized virtual device in a second mode, where in the first mode the bimodal virtual device emulates a real hardware device, and in the second mode the bimodal virtual device functions as an abstract device that is a same type of device as the real hardware device but is incompatible with software configured to interact with the real hardware device.

32.   (Previously presented) The computer-readable instructions of claim 21 further comprising instructions for:
the bimodal virtual device to selectively operate in the first mode when a driver interfacing with said bimodal virtual device has not been designated to interface with said bimodal virtual device operating in said second mode; and
the bimodal virtual device to selectively operate in the second mode when the driver interfacing with said bimodal virtual device has been designated to interface with said bimodal virtual device operating in said second mode.

33.   (Original)   The computer-readable instructions of claim 32 further comprising instructions for the functionality of the second mode to extend the functionality of the first mode.

34.   (Original)   The computer-readable instructions of claim 32 further comprising instructions for the functionality of the second mode that are separate and distinct from instructions for the functionality of the first mode.

35.   (Original)   The computer-readable instructions of claim 34 further comprising instructions for the second mode to disable the functionality of the first mode.

36.   (Original)   The computer-readable instructions of claim 34 further comprising instructions for the second mode to disable portions of the functionality of the first mode.

37.   (Original)   The computer-readable instructions of claim 32 further comprising instructions for enabling the second mode through the use of at least one bit in a virtual device register.

38.   (Original)   The computer-readable instructions of claim 32 further comprising instructions for enabling the second mode through the use of at least one bit in a register specifically created for utilization by one or more virtual devices.

39.   (Original)   The computer-readable instructions of claim 32 further comprising instructions for enabling the second mode through the use of a prescribed sequence of commands or data that change a value in at least one register.

40.   (Original)   The computer-readable instructions of claim 32 further comprising instructions for:

enabling the second mode through the use of a second mode driver installed within a guest operating system environment; and

if the second mode driver is not present, enabling a first mode through the use of a first mode driver.